Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method of managing visibility of GUI components in an application comprising:

providing a user interface of a visibility manager for selection, for each of a plurality of profiles, of one of a plurality of visibility states for each of at least a subset of the GUI components; and

for displaying a user interface screen of the application according to an applied one of the plurality of profiles:

initializing the application, wherein the initializing includes:

starting the application; and

building the user interface screen of the application with all of the GUI components set as visible;

invoking the visibility manager to:

determine, based on the selected visibility states of the applied profile, which of the GUI components of the built user interface screen are to be set as not visible; and

prior to any display of the built user interface screen, revise the built user interface screen based on the determination; and

displaying the revised user interface screen of the application;

wherein the invoking the visibility manager is by:

reading the plurality of profiles,

processing the plurality of profiles,

reading and processing a user configuration based on the plurality of profiles, and

activating the applied profile by:

selecting an identification of a particular GUI component,

locating the identification in a mapping table,

checking a state of the particular ~~GUT~~ GUI component,

comparing the state to the applied profile,

changing the state if not in agreement with the applied profile, and

repeating locating the identification, checking the state of the particular GUI component, comparing the state and changing the state for any remaining additional identifications of additional GUI components.

2-5. (Canceled).

6. (Previously Presented) The method as recited in claim 1 wherein the state is visible or not visible.

7. (Previously Presented) The method as recited in claim 1 wherein the mapping table comprises a plurality of identifications of GUI components and a corresponding plurality of references to objects of an object-oriented and platform independent programming language.

8. (Canceled).

9. (Currently Amended) A system for managing visibility of GUI components in an application comprising:

a processor configured to provide:

a user interface module of the application;

a visibility manager; and

a user interface of the visibility manager via which to receive, for one or more profiles, input of respective selections of visibility states of at least a subset of the GUI components;

wherein: [[,]]

for display of a single user interface screen of the application according to an applied one of the profiles:

the user interface module of the application is configured to initially build the user interface screen with all of the GUI components set as visible and subsequently call the visibility manager;

the visibility manager is configured to be invoked to, in response to the call:

determine, based on the selected visibility states of the at least a subset of the GUI components, which of the GUI components of the built user interface screen are to be set as not visible;

revise the built user interface screen based on the determination, the built user interface screen not being displayed prior to the revision; and

provide the revised user interface screen to the user interface module; and the user interface module is configured to display the revised user interface screen; and

the invocation of ~~invoking~~ the visibility manager [[by]] includes invoking the visibility manager to:

read ~~reading~~ the plurality of profiles,

process ~~processing~~ the plurality of profiles,

read ~~reading~~ and processing a user configuration based on the plurality of profiles, and

activate ~~activating~~ the applied profile by:

selecting an identification of a particular GUI component,

locating the identification in a mapping table,

checking a state of the particular ~~GUT~~ GUI component,

comparing the state to the applied profile,

changing the state if not in agreement with the applied profile, and

repeating locating the identification, checking the state of the particular GUI component, comparing the state and changing the state for any remaining additional identifications of additional GUI components.

10-13. (Canceled).

14. (Previously Presented) The system as recited in claim 9 wherein the state is visible or not visible.

15. (Previously Presented) The system as recited in claim 9 wherein the mapping table comprises a plurality of identifications of GUI components and a corresponding plurality of references to objects of an object-oriented and platform independent programming language.

16. (Canceled).

17. (Currently Amended) A system comprising:

a processor configured to:

use a visibility manager data structure in managing visibility of GUI components in a single user interface screen of an application, the visibility manager data structure comprising a mapping table, one or more profiles and a user configuration identifying which of the one or more profiles is to be applied;

provide a user interface of a visibility manager via which to receive for the one or more profiles input of respective selections of visibility states of at least a subset of the GUI components; and

display the user interface screen of the application according to an applied one of the profiles using the application and the visibility manager, the displaying of the user interface screen including:

the application building the user interface screen of the application with all of the GUI components set as visible and subsequently calling the visibility manager;

the visibility manager, responsive to the calling, being invoked to:

determining, based on the selected visibility states of the applied profile, which of the GUI components of the built interface screen are to be set as not visible; and

prior to any display of the built user interface screen, revising the built user interface screen based on the determination; and

the application displaying the revised user interface screen of the application; [[.]]

wherein the invocation of ~~invoking~~ the visibility manager [[by]] includes invoking the visibility manager to:

read ~~reading~~ the one or more profiles upon initialization of the application,

process ~~processing~~ the plurality of profiles,

read ~~reading~~ and processing a user configuration based on the plurality of profiles, and

activate ~~activating~~ the applied profile by:

selecting an identification of a particular GUI component,

locating the identification in a mapping table,

checking a state of the particular ~~GUT~~ GUI component,

comparing the state to the applied profile,

changing the state if not in agreement with the applied profile, and repeating locating the identification, checking the state of the particular GUI component, comparing the state and changing the state for any remaining additional identifications of additional GUI components.

18-21. (Canceled).

22. (Previously Presented) The system as recited in claim 17 wherein the state is visible or not visible.

23. (Previously Presented) The system as recited in claim 17 wherein the mapping table comprises a plurality of identifications of GUI components and a corresponding plurality of references to objects of an object-oriented and platform independent programming language.

24. (Canceled).

25. (Currently Amended) A hardware-implemented computer-readable medium embodying instructions, which, when executed by a processor, cause the processor to perform a method, the method comprising:

providing a user interface of a visibility manager via which to receive, for one or more profiles, input of respective selections of visibility states of at least a subset of GUI components; and

for displaying a user interface screen of the application according to an applied one of the plurality of profiles:

initializing an application, wherein the initializing includes:

starting the application; and

building the user interface screen of the application with all of the GUI components set as visible;

invoking a visibility manager to:

determine, based on the selected visibility states of the applied profile, which of the GUI components of the built user interface screen are to be set as not visible; and

prior to any display of the built user interface screen, revise the built user interface screen based on the determination; and displaying the revised user interface screen of the application; [[.]]

wherein the invoking the visibility manager is by:

reading the one or more profiles,

processing the one or more profiles,

reading and processing a user configuration based on the one or more profiles, and

activating a particular profile of the one or more profiles by:

selecting an identification of a particular GUI component,

locating the identification in a mapping table,

checking a state of the particular GUI component,

changing the state if not in agreement with the particular profile of the one or more profiles, and

repeating locating the identification, checking the state of the particular GUI component, comparing the state and changing the state for any remaining additional identifications of additional GUI components.

26-29. (Canceled).

30. (Previously Presented) The medium as recited in claim 25 wherein the state is visible or not visible.

31. (Previously Presented) The medium as recited in claim 25 wherein the mapping table comprises a plurality of identifications of GUI components and a corresponding plurality of references to objects of an object-oriented and platform independent programming language.

33. (Currently Amended) A system for managing visibility of GUI components in an application, the system implemented via hardware components including a processor, comprising:

first means for interfacing with a user, the means for interfacing providing the GUI components for display; and

means for determining which GUI components are visible, the means for determining including second means for interfacing with a user, the second means for interfacing providing for receipt, for one or more profiles, of respective user selections of visibility states of at least a subset of the GUI components;

wherein: [[,]]

for display of a user interface screen of the application according to an applied one of the profiles:

the first means for interfacing is configured to initially build the user interface with all of the GUI components set as visible and subsequently call the means for determining;

the means for determining is configured to, in response to the call:

determine, based on the selected visibility states of the at least a subset of the GUI components, which of the GUI components of the built user interface screen are to be set as not visible;

prior to any display of the built user interface screen, revise the built user interface screen based on the determination; and

provide the revised user interface screen to the user interface module; and

the first means is configured to display the revised user interface screen; and ~~. wherein~~

the means for determining includes:

means for reading the one or more profiles upon initialization of the application;

means for processing the one or more profiles;

means for reading and processing a user configuration based on the one or more profiles; and

means for activating a particular profile of the one or more profiles by:

selecting an identification of a particular GUI component,

locating the identification in a mapping table,

checking a state of the particular GUI component, changing the state if not in agreement with the particular profile of the one or more profiles, and

repeating locating the identification, checking the state of the particular GUI component, comparing the state and changing the state for any remaining additional identifications of additional GUI components.

34-35. (Canceled).

36. (Currently Amended) A method for displaying a user interface, comprising:

starting up a visibility manager;

an application building a data structure representing a user interface including a plurality of interface components;

during the building of the data structure and responsive to the start-up of the visibility manager, the visibility manager selecting one of a plurality of interface profiles, each of the interface profiles indicating for each of a plurality of interface components a respective visibility instruction;

subsequent to the building of the data structure, the application calling the visibility manager;

responsive to the call and prior to any display of the user interface, the visibility manager modifying the data structure in accordance with the visibility instructions of the selected interface profile, the modified data structure representing a modified version of the user interface which does not include all of the plurality of interface components; and

the application displaying the modified version of the user interface; [[.]]

wherein ~~invoking~~ the modifying of the data structure by the visibility manager [[by]] includes:

reading the plurality of interface profiles upon initialization of the application,

processing the plurality of interface profiles,

reading and processing a user configuration based on the plurality of interface profiles, and

activating the selected interface profile by:

selecting an identification of a particular GUI component,

locating the identification in a mapping table, checking a state of the particular GUI component,

comparing the state to the selected interface profile,

changing the state if not in agreement with the selected interface profile, and

repeating locating the identification, checking the state of the particular GUI component, comparing the state and changing the state for any remaining additional identification of additional GUI components.

37. (Previously Presented) The method of claim 36, wherein the starting up of the visibility manager and the building of the data structure by the application are performed in response to a start-up of the application.

38. (Previously Presented) The method of claim 36, wherein the selection of the selected interface profile is based on a user type associated with a present use of the application.

39. (Previously Presented) The method of claim 36, wherein the selection of the selected interface profile is based on a user configuration file which identifies which of the plurality of interface profiles a user associated with a present use of the application has set as active.